|  |  |  |
| --- | --- | --- |
| МИНОБРНАУКИ РОССИИ | МИНОБРНАУКИ РОССИИ | МИНОБРНАУКИ РОССИИ |
| *Федеральное государственное бюджетное образовательное учреждение*  *высшего образования*  ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** | *овательное учреждение*  *высшего образования*  ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** | *Федеральное государственное бюджетное образовательное учреждение*  *высшего образования*  ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** |



Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации информационных технологий (МОСИТ)

ПРАКТИЧЕСКАЯ РАБОТА №1

по дисциплине

# «Тестирование и верификация программного обеспечения»

|  |  |
| --- | --- |
| **Выполнили:**  Студенты группы ИКБО-36-22 | Аскаров Т. Е.  Захаров А. А.  Ярош В. Э.  Утенков Ю. Ю.  . |
| **Проверил:** | ассистент Петрова А. А. |

Москва 2024 г.

СОДЕРЖАНИЕ

[«Тестирование и верификация программного обеспечения» 1](#_Toc176890039)

[ФОРМУЛИРОВКА ЗАДАНИЯ 4](#_Toc176890040)

[ВЫПОЛНЕНИЕ ЗАДАНИЯ 6](#_Toc176890041)

[Техническое задание на разработку программного погодного телеграмм бота. 6](#_Toc176890042)

[1. Введение 6](#_Toc176890043)

[2. Общие сведения 6](#_Toc176890044)

[2.1 Назначение 6](#_Toc176890045)

[2.2 Краткий обзор 6](#_Toc176890046)

[3. Назначение и цели создания системы 7](#_Toc176890047)

[3.1 Назначение системы 7](#_Toc176890048)

[3.2 Цели создания системы 8](#_Toc176890049)

[4. ХАРАКТЕРИСТИКА ОБЪЕКТА АВТОМАТИЗАЦИИ 9](#_Toc176890050)

[4.1.1 Объект автоматизации 9](#_Toc176890051)

[4.1.2 Условия эксплуатации и характеристики среды 9](#_Toc176890052)

[5. Требования к автоматизированной системе 10](#_Toc176890053)

[5.1 Функциональные требования 10](#_Toc176890054)

[5.1.1 Управление рассылочными данными 10](#_Toc176890055)

[5.1.2 Карты температурных зон 10](#_Toc176890056)

[5.1.3 Выбор требуемого языка 10](#_Toc176890057)

[5.1.5 Просмотр прогноза на несколько дней вперёд 11](#_Toc176890058)

[5.2 Нефункциональные требования 11](#_Toc176890059)

[5.2.1 Надежность 11](#_Toc176890060)

[5.2.2 Доступность 11](#_Toc176890061)

[5.2.3 Безопасность 11](#_Toc176890062)

[6. Состав и содержание работ по созданию (развитию) системы 11](#_Toc176890063)

[7. Порядок разработки автоматизированной системы 12](#_Toc176890064)

[8. Порядок контроля и приемки автоматизированной системы 13](#_Toc176890065)

[9. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие 13](#_Toc176890066)

[10. Требования к документации 13](#_Toc176890067)

[11. Источники разработки 13](#_Toc176890068)

[12. Ошибки, заложенные в программном продукте 14](#_Toc176890069)

[ПРОВЕРКА ТЗ И ДОКУМЕНТАЦИИ ПРОГРАММНОГО ПРОДУКТА ДРУГОЙ КОМАНДЫ 17](#_Toc176890070)

[Отчёт по тестированию 28](#_Toc176890071)

[Приложение 29](#_Toc176890072)

[Вывод 33](#_Toc176890073)

[ЗАКЛЮЧЕНИЕ 34](#_Toc176890074)

# ФОРМУЛИРОВКА ЗАДАНИЯ

**Часть 1**

1. Для выполнения задания, группа должна разделится на команды численность 2-4 человека и придумать себе название. Команда выбирается на весь курс дисциплины.
2. Каждая из команд для выполнения 1 части практического задания подготавливает следующие материалы:

* рассматриваемый Программный продукт;
* техническое задание под рассматриваемый программный продукт и дополнительную документацию при необходимости.

Спецификации на предыдущие материалы:

* программный продукт должен быть исполняемым, удобным и читаемым для выполнения другой командой;
* должно присутствовать полное и достаточное описание для запуска программного продукта в случае его нетривиального исполнения;
* программный продукт должен содержать в себе определенное количество ошибок при исполнении от 5-8;
* техническое задание должно полностью описывать функциональные возможности программного продукта;
* описание в каждом пункте технического задания должно быть кратким, лаконичным и не вызывать двояких трактовок;
* техническое задание составляется по шаблону.

**Часть 2**

На момент выполнения второй части практического задания у команд в группе должно быть выполнена часть 1 практического задания.

1. Команда берет программный продукт, созданный другой командой.
2. Каждая команда изучает программный продукт, выбранный для тестирования, изучает ТЗ, документацию.
3. Анализирует полноту и качество описания ПП в ТЗ и документации.
4. Команда должна протестировать ПП, используя метод «черного ящика».
5. Исходя из конечного результата тестирования командой составляется отчет, по проделанной работе.
6. Командная работа подразумевает четкое разделение ролей. При необходимости каждый студент должен объяснить свою роль, в выполнении практической работы.

# ВЫПОЛНЕНИЕ ЗАДАНИЯ

Техническое задание на разработку программного погодного телеграмм бота.

1. Введение

Тестирование программного обеспечения охватывает множество аспектов, включая функциональное, нагрузочное и регрессионное тестирование. Каждый из этих типов тестирования имеет свои цели и методы, что позволяет выявить различные виды дефектов на разных этапах разработки. Например, функциональное тестирование направлено на проверку корректности работы всех функций программы и её соответствие заданным требованиям, в то время как нагрузочное тестирование помогает оценить производительность системы при высоких нагрузках. Современные инструменты автоматизации тестирования значительно повышают эффективность процесса проверки. Использование автоматизированных тестов позволяет не только ускорить выявление ошибок, но и снизить вероятность человеческого фактора, который может привести к пропускам критически важных дефектов. Однако, несмотря на все преимущества автоматизации, ручное тестирование по-прежнему остается важной частью QA-процесса, особенно в случаях, когда необходима интуитивная оценка пользовательского опыта.

1. Общие сведения
   1. Назначение

Программное приложение погодный телеграмм бот.

* 1. Краткий обзор

Погодный телеграмм бот на Java Spring Boot – это приложение с визуальным интерфейсом телеграмма, позволяющее любому пользователю узнавать погоду по всему миру с возможностью прогноза на 5 дней вперёд. При помощи визуального интерфейса пользователь может нажимать на различные кнопки, например: выбор языка общения, выбор дня просмотра погоды, показ выпадающего списка с доступными командами бота. Также пользователь имеет возможность просматривать температурную карту. Приложение создано для пользователей, умеющих обращаться с мессенджером телеграмм. Помимо этого, приложение должно быть стабильным и высокопроизводительным. Также для пользователей с ролью ADMIN доступна ещё одна команда **/send** с возможностью рассылки сообщений пользователям.

* 1. **Разработчики**

Состав команды: Аскаров Т. Е., Захаров А. А., Ярош В. Э., Утенков Ю. Ю..

* 1. **Заказчики**

Казаков И. Г., Кужугет С. А., Соколов Р. В., Шурша И. А.

* 1. **Основание для разработки**

Договор № 123.45 от 06.09.2024 на разработку автоматизированной системы.

1. Назначение и цели создания системы
   1. Назначение системы

Телеграм-бот для прогноза погоды предназначен для предоставления пользователям актуальной метеорологической информации и упрощения доступа к прогнозам погоды. Основные функции, которые предоставляет бот, включают:

1. Поддержка нескольких языков: Бот поддерживает 4 языка (русский, английский, немецкий и китайский), что позволяет пользователям получать информацию на удобном для них языке.

2. Прогноз погоды: Бот предоставляет прогноз погоды на текущий день и на несколько дней вперед, включая данные о температуре, осадках, ветре и уровне влажности.

3. Температурная карта: Для более наглядного представления бот отображает температурную карту, которая визуализирует текущие температуры в различных регионах.

4. Рассылка уведомлений: Пользователи с ролью ADMIN могут рассылать полезную информацию всем остальным пользователям.

Бот предназначен для удобного и быстрого получения информации о погоде и позволяет пользователям быть в курсе текущих и будущих метеоусловий, обеспечивая простоту использования и точность данных.

* 1. Цели создания системы

Целями создания телеграм-бота для прогноза погоды являются:

* Обеспечение пользователей быстрым и удобным доступом к информации о текущей погоде и прогнозе на ближайшие дни.
* Автоматизация процесса получения погодных данных, что позволит пользователям сэкономить время и минимизировать необходимость поиска информации вручную.
* Повышение точности и актуальности предоставляемой информации за счет использования данных от проверенных метеорологических источников.
* Обеспечение возможности для пользователей получать предупреждения о резких изменениях погоды или неблагоприятных условиях.
* Создание системы, которая поддерживает несколько языков, что расширит круг пользователей и сделает сервис доступным для более широкой аудитории.

Для достижения этих целей бот должен:

* Иметь простой и интуитивно понятный интерфейс, адаптированный под мобильные устройства.
* Быть высокопроизводительным и надежным при запросах большого числа пользователей.
* Поддерживать интеграцию с погодными API для получения и обработки данных в режиме реального времени.
* Обеспечивать возможность настройки уведомлений и рассылки прогнозов по запросам пользователей.
* Поддерживать многоязычность и возможность масштабирования для добавления новых функций в будущем.

1. ХАРАКТЕРИСТИКА ОБЪЕКТА АВТОМАТИЗАЦИИ
   * 1. Объект автоматизации

Объектом автоматизации является процесс предоставления погодной информации пользователям, включая получение и отображение прогноза, рассылку уведомлений, поддержку многоязычного интерфейса и управление запросами. Телеграм-бот предназначен для упрощения взаимодействия пользователей с метеорологическими данными, обеспечивая удобный доступ к актуальной информации о погоде и повышая эффективность процесса её получения и использования.

* + 1. Условия эксплуатации и характеристики среды

Программное обеспечение будет эксплуатироваться в условиях домашней среды с доступом к интернету. Клиенты будут взаимодействовать с системой через телеграмм-интерфейс, что требует наличия современного электронного устройства. Система разрабатывается на основе технологии Java Spring Boot и будет использовать базу данных PostgreSQL для хранения данных.

Система рассчитана на эксплуатацию в стандартных домашних, без особых требований к окружающей среде. Основные требования включают:

* Наличие стабильного интернет-соединения для доступа к веб-интерфейсу системы.
* Работа на серверном оборудовании, соответствующем требованиям для запуска приложений на платформе Java Spring Boot.
* Совместимость с операционными системами, поддерживающими Java.

1. Требования к автоматизированной системе
   1. Функциональные требования
      1. Управление рассылочными данными

* Администратор должен иметь возможность добавлять необходимую информацию с возможностью рассылки её остальным пользователям для ознакомления.
* Клиент должен иметь возможность просмотра текущих данных, отправленных администратором.
  + 1. Карты температурных зон
* При вводе команды /map пользователи имеют возможность получить текущую температурную карту для выбранного региона.
  + 1. Выбор требуемого языка
* При вводе команды /lang пользователи имеют возможность выбрать требуемый язык для удобного общения с ботом. Вся присланная ботом информация будет на выбранном языке.
  + 1. **Просмотр прогноза погоды**
* При успешном выборе языка, пользователи могут уже работать с ботом, и отправлять ему сообщением город для получения погодных данных о нём на выбранном языке.
  + 1. Просмотр прогноза на несколько дней вперёд
* При успешном выборе языка, пользователи могут отправлять ему сообщением, в формате <Город> <количество дней для прогноза> для получения погодных данных о нём на выбранном языке на заданное количество дней вперёд.
  1. Нефункциональные требования
     1. Надежность

Приложение должно работать при любых настройках системы, на которой оно запускается. Помимо этого, программное приложение должно поддерживать все основные видеоформаты, также оно должно поддерживать различные разрешения монитора компьютера.

* + 1. Доступность

Приложение должно быть готовым к работе при любом состояние системы, а также оно должно быть полностью автономным.

* + 1. Безопасность

В данномпрограммном приложении изначальная роль нового пользователя НЕ должна быть ADMIN, все новые пользователи должны обладать ролью CLIENT.

1. Состав и содержание работ по созданию (развитию) системы

* Разработка технического задания:
* Содержание: Определение требований, создание и согласование технического задания.
* Результат: Утвержденное техническое задание.
* Проектирование системы:
* Содержание: Разработка архитектуры системы, проектирование базы данных и интерфейсов.
* Результат: Документация по архитектуре, проект базы данных, макеты интерфейсов.
* Разработка программного обеспечения:
* Содержание: Написание и тестирование кода.
* Результат: Рабочее программное обеспечение.
* Интеграция и тестирование:
* Содержание: Интеграция модулей, функциональное и нагрузочное тестирование.
* Результат: Протестированная система.
* Внедрение системы:
  + Содержание: Установка, настройка, обучение пользователей.
  + Результат: Рабочая система и обученные пользователи.
* Поддержка и сопровождение:
* Содержание: Техническая поддержка, обновления.
* Результат: Стабильная работающая система.

1. Порядок разработки автоматизированной системы

Процесс создания системы проходит через несколько ключевых фаз: анализ требований, проектирование, программирование, тестирование и внедрение. Перед началом каждого нового этапа требуется получение одобрения от заказчика. Важным аспектом являются промежуточные проверки для обеспечения соответствия системы установленным требованиям.

План разработки согласуется с ассистентом кафедры МОСИТ Петровой А.А.

Необходимые этапы при разработки автоматизированной системы: аналитический, проектный, программный, тестирование и внедрение.

1. Порядок контроля и приемки автоматизированной системы

Контроль и приемка системы включают функциональные, интеграционные и нагрузочные испытания, проводимые в соответствии с действующими стандартами. Испытания охватывают проверку всех компонентов системы на соответствие требованиям технического задания. Приемка выполняется поэтапно: документация проверяется и утверждается в установленные сроки, а участие в приемке принимает заказчик, разработчик и, при необходимости, сторонние эксперты. Приемочная комиссия, состоящая из представителей заказчика и разработчика, утверждает результаты и принимает решение о готовности системы к эксплуатации.

1. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие

Для подготовки объекта автоматизации к внедрению системы необходимо выполнить ряд ключевых мероприятий. Это включает в себя: подготовку информации для обработки в системе, внесение изменений в объект автоматизации, обеспечение соответствующих условий для функционирования системы, создание необходимых служб и подразделений, а также организацию обучения персонала. Эти мероприятия должны быть согласованы с заказчиком и выполнены до начала эксплуатации системы, а их детали уточняются на стадии разработки и опытной эксплуатации.

1. Требования к документации

Для автоматизированной системы требуется разработка и поддержка следующих документов: техническое задание, проектная документация, руководство пользователя и эксплуатационная документация. Все документы должны соответствовать стандартам ГОСТ и внутренним нормативам. Документация будет предоставляться как в бумажном, так и в электронном виде.

1. Источники разработки

* Договор № 123.45 от 06.09.2024;
* ГОСТ 34.602 – 2020 "Техническое задание на создание автоматизированной системы".

1. Ошибки, заложенные в программном продукте

Подразумевается, что прочтение данного раздела происходит после ознакомления с документацией и пониманием основных функций приложения.

При запуске приложения, если пропустить выбор языка, то по умолчанию будет общение на немецком, хотя должно быть на английском (рисунок 1).

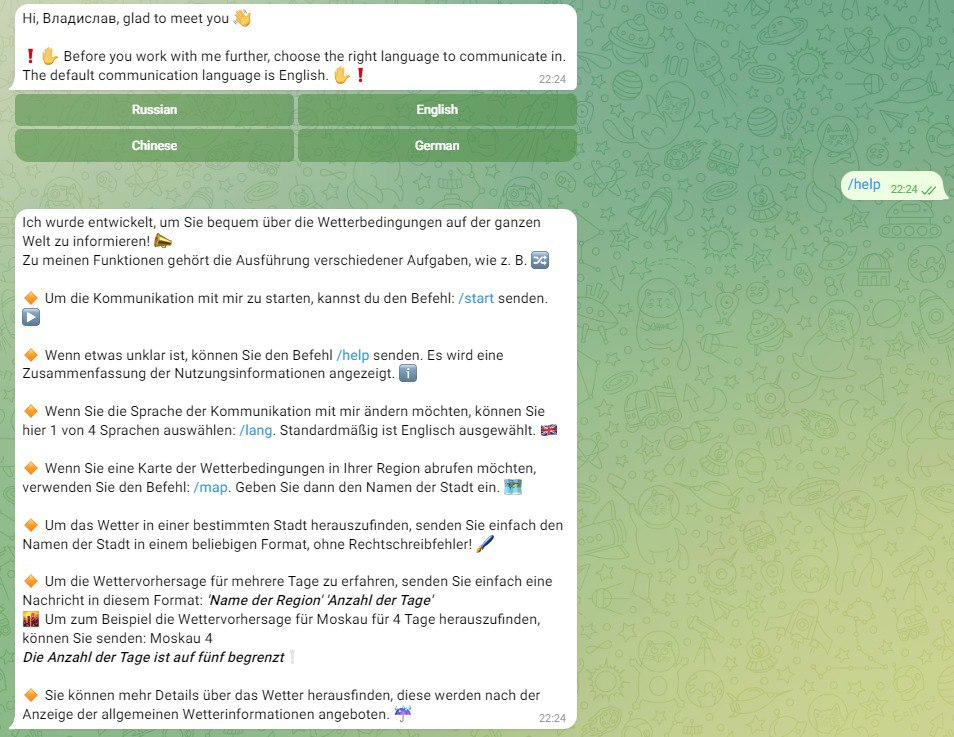


Рисунок 1 – Пример некорректной работы

При начале общения нового пользователя, данный пользователь сохраняется в базу данных с ролью ADMIN, хотя должен сохраняться с ролью CLIENT (рисунок 2).

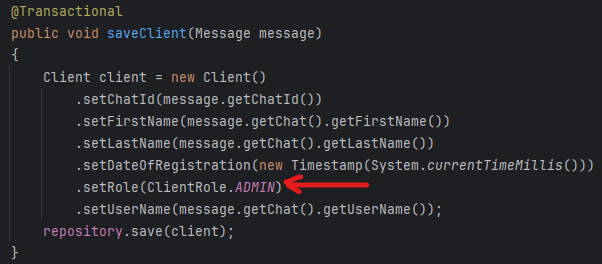


Рисунок 2 – Отображение ошибки в программном коде

Сообщение “Подробный прогноз” для китайского языка в действительности написано на русском (рисунок 3).

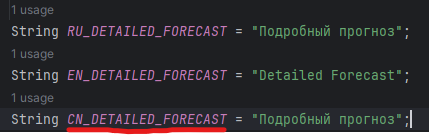


Рисунок 3 – Отображение ошибки в программном коде

При первом запуске бота, сообщение с просьбой ввести команду **/star**, а в действительности нужно ввести команду **/start**,из-за чего команда не будет распознана ботом (рисунок 4).



Рисунок 4 – Отображение ошибки в программном коде

При возникновении ошибки с выбранным русским языком, сообщение об ошибке будет выведено на английском языке, вместо русского (рисунок 5).

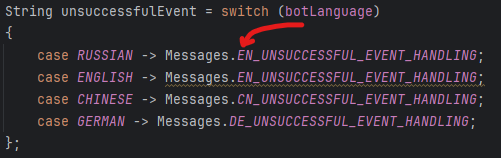


Рисунок 5 – Отображение ошибки в программном коде

ПРОВЕРКА ТЗ И ДОКУМЕНТАЦИИ ПРОГРАММНОГО ПРОДУКТА ДРУГОЙ КОМАНДЫ

1. ОБЩИЕ СВЕДЕНИЯ

1.1 Полное наименование системы и ее условное обозначение

Полное наименование системы: Интегрированная многопользовательская облачная платформа для хранения и совместной работы с документами «Cloud file storage».

Краткое наименование системы: Cloud file storage.

1.2 Номер договора (контракта)

Наличие договора не предусмотрено.

1.3 Наименования организации-заказчика и организаций-участников работ

Заказчиком системы является Лимонов А.А.

Адрес заказчика: 222000 г. Москва, Лубянка, д.1.

Разработчиком системы является Лимонов А.А.

Адрес разработчика: 222000 г. Москва, Лубянка, д.1.

1.4 Перечень документов, на основании которых создается система

Документы, на основании которых создаётся система, не предусмотрены.

1.5 Плановые сроки начала и окончания работы по созданию системы

Плановый срок начала работ по созданию Интегрированной многопользовательской облачной платформы для хранения и совместной работы с документами «Cloud file storage» – 20 августа 2024 года.

Плановый срок окончания работ по созданию Интегрированной многопользовательской облачной платформы для хранения и совместной работы с документами «Cloud file storage» – 06 сентября 2024 года.

1.6 Источники и порядок финансирования работ

Источники финансирования отсутствуют.

1.7 Порядок оформления и предъявления заказчику результатов работ по созданию системы

Система предоставляется в виде полностью собранного и протестированного комплекса на основе современной вычислительной техники. Проверка и приемка системы осуществляется Заказчиком после получения.

1.8 Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ

При создании проектно-эксплуатационной документации Исполнитель должен руководствоваться требованиями следующих нормативных документов:

– ГОСТ 34.602-2020. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания

1.9 Определения, обозначения и сокращения

Определения, обозначения и сокращения отсутствуют.

2. НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ

2.1 Назначение системы

Система «Cloud file storage» предназначена для автоматизации процессов управления пользовательскими файлами. Основные функции системы включают:

- регистрация и авторизация пользователей,

- загрузка и хранение файлов пользователей,

- возможность создания новых папок пользователем,

- возможность удаления файлов и папок пользователем.

Система не предполагает использования на объектах автоматизации.

2.2 Цели создания системы

«Cloud file storage» создаётся в следующих целях:

- Разработка многопользовательского файлового облака с возможностью загрузки и хранения файлов.

- Практическое применение технологий Spring Boot, Docker и NoSQL хранилищ для создания современного веб-приложения.

- Практика в реализация функционала работы с пользователями (авторизация, регистрация) и файлами (загрузка, удаление, переименование).

3. ХАРАКТЕРИСТИКА ОБЪЕКТА АВТОМАТИЗАЦИИ

Объектами автоматизации являются процессы управления пользователями, файлами и папками в многопользовательской системе хранения данных, аналогичной Google Drive, с использованием облачных технологий для хранения и обработки информации.

Процессы автоматизации включают:

- Регистрация и авторизация пользователей

- Управление файлами и папками

- Взаимодействие с S3-совместимым хранилищем

- Интерфейс поиска файлов и папок

- Управление доступом и сессиями

- Формирование и обработка HTTP-запросов

- Формирование отчётов и логов

- Администрирование и поддержка пользователей

Эти процессы обеспечивают полный цикл работы облачного хранилища, включая управление данными, пользовательский доступ и взаимодействие с облачной инфраструктурой.

4. ТРЕБОВАНИЯ К СИСТЕМЕ

4.1 Требования к системе в целом

4.1.1 Требования к структуре и функционированию системы

4.1.1.1 Перечень подсистем, их назначение и основные характеристики

Наличие подсистем не предусмотрено в составе «Cloud file storage».

4.1.1.2 Требования к способам и средствам связи для информационного обмена между

Сохранение данных (реализованное через базу данных) и обработка данных внутри модуля обработки и хранения информации должны взаимодействовать друг с другом, используя сетевые протоколы TCP/IP.

Клиентское приложение (например, веб-браузер) и веб-сервер должны обмениваться данными по стандартным сетевым протоколам HTTP/HTTPS.

4.1.1.3 Требования к характеристикам взаимосвязей создаваемой системы со смежными

Требования не предъявляются.

4.1.1.4 Требования к режимам функционирования системы

Требования не предъявляются.

4.1.1.5 Требования по диагностированию системы

Требования не предъявляются.

4.1.1.6 Перспективы развития, модернизации системы

Проектные решения в Системе должны обеспечивать возможность дальнейшего развития и модернизации, расширение функциональных возможностей за счет дополнительной разработки и внедрения новых модулей.

Система должна быть способна к росту производительности и увеличению объема хранимых данных без необходимости изменения ПО, путем масштабирования комплекса технических средств системы.

4.1.2 Требования к численности и квалификации персонала системы

Требования не предъявляются.

4.1.3 Показатели назначения

Система должна обеспечивать параллельную работу не менее 1000 авторизованных пользователей. Ожидаемое время отклика при доступе к графическим интерфейсам не должно превышать 10 секунд. Система должна обеспечивать горизонтальное масштабирование производительности, включая возможность увеличения числа пользователей сверх расчетного значения, без изменения программного обеспечения путем обновления используемых технических средств.

4.1.4 Требования к надежности

Композиция элементов и структура системы должны гарантировать коэффициент использования ресурсов не ниже 99.7%. Под этим понимается степень готовности информационной инфраструктуры, при которой субъекты с правами доступа могут свободно осуществлять свои действия без преград.

Этот уровень эффективности означает, что система работает практически непрерывно, с минимальными периодами простоя или снижения производительности.

4.1.5 Требования к безопасности

Требования не предъявляются.

4.1.6 Требования к эргономике и технической эстетике

Требования не предъявляются.

4.1.7 Требования к транспортабельности для подвижных АС

Требования не предъявляются.

4.1.8 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Требования не предъявляются.

4.1.9 Требования к защите информации от несанкционированного доступа

Компоненты защиты от НСД должны обеспечивать:

– идентификацию пользователя;

– проверку полномочий пользователя при работе с системой.

4.1.10 Требования по сохранности информации при авариях

Требования не предъявляются.

4.1.11 Требования к защите от влияния внешних воздействий

Требования не предъявляются.

4.1.12 Требования к патентной чистоте

Требования не предъявляются.

4.1.13 Требования по стандартизации и унификации

Требования не предъявляются.

4.2 Требования к функциям (задачам), выполняемым системой

Модуль управления пользователями должен обеспечивать выполнение следующих функций:

- Регистрация: Пользователь должен иметь возможность зарегистрироваться в системе, предоставив уникальные учетные данные.

- Авторизация: Авторизация должна позволять пользователям входить в систему с помощью предоставленных учетных данных.

- Выход из аккаунта: Реализовать функцию выхода из системы с возможностью завершения текущей сессии пользователя.

Требования к качеству реализации:

- Регистрация и авторизация должны обеспечивать безопасность данных пользователя, используя шифрование паролей.

- Система должна поддерживать одновременную работу нескольких пользователей.

- Время на выполнение регистрации и авторизации не должно превышать 2 секунд.

Модуль управления файлами и папками должен включать в себя следующие модули:

- Загрузка файлов и папок: Пользователь должен иметь возможность загружать файлы и папки любого типа.

- Создание новой папки: Возможность создания новой пустой папки аналогично созданию папок в файловом менеджере.

- Удаление: Пользователь должен иметь возможность удалять файлы и папки.

- Переименование: Функция переименования файлов и папок.

Требования к качеству реализации:

- Все операции с файлами и папками должны выполняться в течение 1-3 секунд для файлов размером до 100 МБ.

- Система должна гарантировать, что при удалении данных они могут быть восстановлены в течение 30 дней.

- Функция переименования должна автоматически проверять наличие дубликатов имен.

Модуль хранения данных отвечает за хранение загруженных пользователями файлов и данных системы:

- Хранение файлов в облачном S3-совместимом хранилище.

- Обеспечение резервного копирования данных для защиты от потерь.

Требования к качеству:

- Обеспечить безопасность данных за счет шифрования при хранении и передаче.

4.3 Требования к видам обеспечения

4.3.1 Требования к математическому обеспечению системы

Методологии и процедуры, применяемые для кодирования и декодирования информации, а также программные инструменты, реализующие эти процессы, должны пройти сертификацию уставшими органами для применения в государственных учреждениях Российской Федерации.

4.3.2 Требования к информационному обеспечению системы

Нивелировка данных в системе должна основываться на актуальных реляционных или объектно-реляционных системах управления базами данных (СУБД). Для обеспечения целостности информации используются встроенные механизмы этих систем. Инструменты СУБД и функции используемых операционных систем должны обеспечивать документацию и логирование обрабатываемой в системе информации.

4.3.3 Требования к лингвистическому обеспечению системы

Все прикладное программное обеспечение системы для организации взаимодействия с

пользователем должно использовать английский язык.

4.3.4 Требования к программному обеспечению системы

Используемое при разработке программное обеспечение и библиотеки программных кодов должны иметь широкое распространение, быть общедоступными и использоваться в промышленных масштабах. Базовой программной платформой должна являться операционная система MS Windows.

4.3.5 Требования к техническому обеспечению

Требования отсутствуют.

5. СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ (РАЗВИТИЮ) СИСТЕМЫ

Состав и содержание работ по подготовке объекта автоматизации к вводу Системы в действие включают этапы:

- Разработка технического задания

- Техническое проектирование (разработка структуры, состава и функций Системы)

- Разработка модулей

- Предварительные испытания (разработка программы и методики испытаний, проведение предварительных испытаний, устранение замечаний, выявленных по результатам предварительных испытаний)

- Опытная эксплуатация (проведение опытной эксплуатации, устранение замечаний, выявленных по результатам приемочных испытаний)

6. ПОРЯДОК РАЗРАБОТКИ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ

Порядок разработки автоматизированной системы включает этапы:

1. Подготовительный этап

1.1. Анализ требований и проектирование: Определение функциональных требований системы, таких как регистрация, авторизация, управление файлами и папками, их взаимодействие. 1.2. Выбор технологий и инструментов: Использование Docker Compose, Spring Boot, Thymeleaf, Minio SDK, MySQL, Redis для реализации системы.

2. Основной этап разработки

2.1. Реализация регистрации и авторизации пользователей:

Внедрение Spring Boot, Spring Security, Thymeleaf и Spring Data JPA для реализации функционала регистрации и авторизации пользователей.

Проведение интеграционных тестов для регистрации.

2.2. Интеграция с MinIO:

Интеграция Minio Java SDK с Spring Boot для выполнения операций с файлами в бакете.

Создание сервисов для работы с файлами и папками.

2.3. Реализация поиска файлов:

Разработка сервисов и контроллеров для поиска файлов с использованием Thymeleaf для отображения результатов.

2.4. Тестирование:

Проведение интеграционных тестов для сервисов работы с файлами и папками.

3. Завершающий этап

3.1. Развертывание:

Завершение настройки системы, подготовка к развертыванию на целевом окружении.

Тестирование стабильности и производительности системы на финальном этапе.

7. ТРЕБОВАНИЯ К СОСТАВУ И СОДЕРЖАНИЮ РАБОТ ПО ПОДГОТОВКЕ ОБЪЕКТА АВТОМАТИЗАЦИИ К ВВОДУ СИСТЕМЫ В ДЕЙСТВИЕ

Требования отсутствуют.

8. ТРЕБОВАНИЯ К ДОКУМЕНТИРОВАНИЮ

Требования отсутствуют.

9. ИСТОЧНИКИ РАЗРАБОТКИ

Источники, на основании которых разрабатывалось ТЗ и которые

должны быть использованы при создании системы:

- «Spring Boot в действии»/ Крейг Уоллс

- «Глубокое погружение в Docker»/ Найджел Поултон

- «Использование Java Persistence с Hibernate»/Кристиан Бауэр, Гэвин Кинг.

### Отчёт по тес**тир**ованию

Выявленные ошибки:

|  |  |  |
| --- | --- | --- |
| Входные значения | Ожидаемый результат | Полученные значения |
| Вводится в URL переменная HTTP запроса path со значением / | Обработка данного параметра в программе и переход по указанному пути | Появляется ошибка со статусом 500 Whitelabel Error Page. Рисунок 6 |
| Создаётся пустая папка в браузере с выбранным названием | Создание пустой папки | Создаётся папка с названием и пустым файлом с таким же названием внутри неё. Рисунок 7 |
| Скачивание папки на локальный компьютер | Папка и все файлы внутри неё рекурсивно копируются на локальный компьютер | При попытке скачать папку скачать её не удаётся. Рисунок 8 |
| Загрузка с локального компьютера выбранной пустой папки на облако | Загрузка пустой папки на облако | Загрузка не происходит, в некоторых случаях появляется неопознанный файл. Рисунок 9 |
| Переименование файла на облачном хранилище | Выбранному объекту успешно задаётся данное название | Если есть 2 файла, то если переименовать один файл в другой, то первый файл исчезает и затеряется. Рисунок 10 |
| Просмотр занятого места на облачном хранилище | Успешный просмотр занятого места на облаке | Занятый объём не отображается пользователю. Рисунок 11 |
| Удаление и редактирование файла | Успешное удаление и редактирование | Невозможность удаления битого файла. Рисунок 12 |
| Удаление и редактирование файла внутри директорий | Корректное удаление файлов внутри директории | При попытке удаления последнего файла внутри директории появляется ошибка Whitelabel Error Page со статусом 500. Рисунок 13 |

### Приложение

Рисунки с ошибками протестированного приложения «Cloud File Storage».

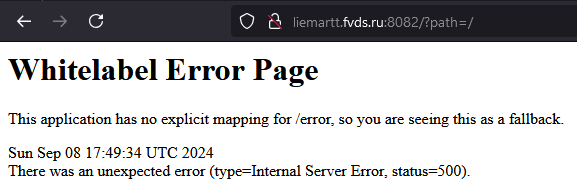


Рисунок 6 – Окно с ошибкой при передаче параметра path=/

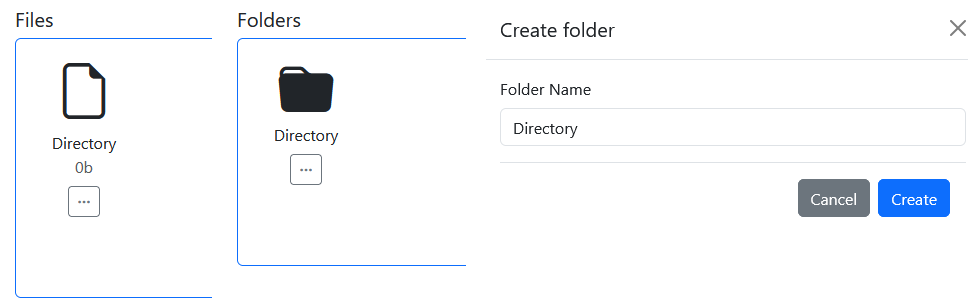


Рисунок 7 – Создание с пустой папки с файлом внутри

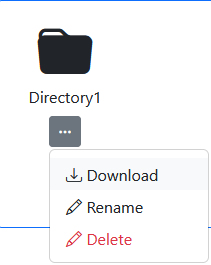


Рисунок 8 – Попытка скачивания директории на компьютер

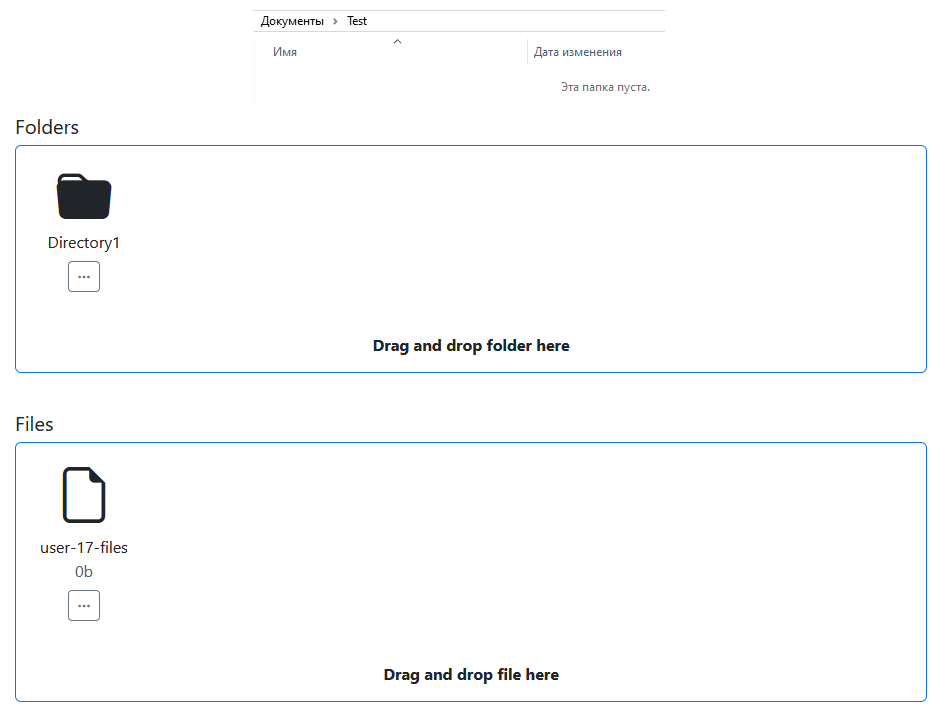


Рисунок 9 – Попытка загрузки пустой директории и результат

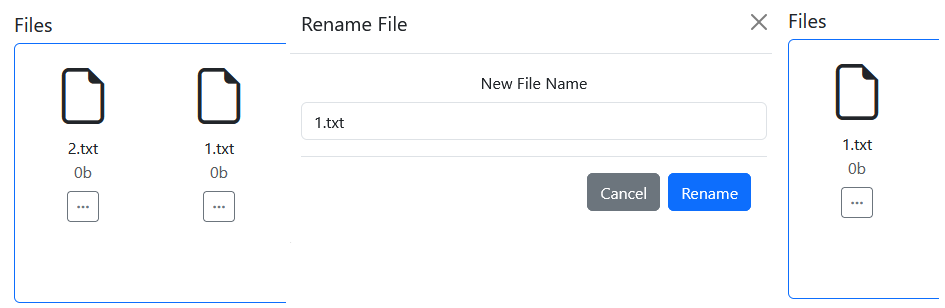


Рисунок 10 – Ошибки с переименованием файла

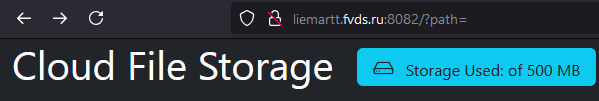


Рисунок 11 – Не отображается занятый объём данных на облаке

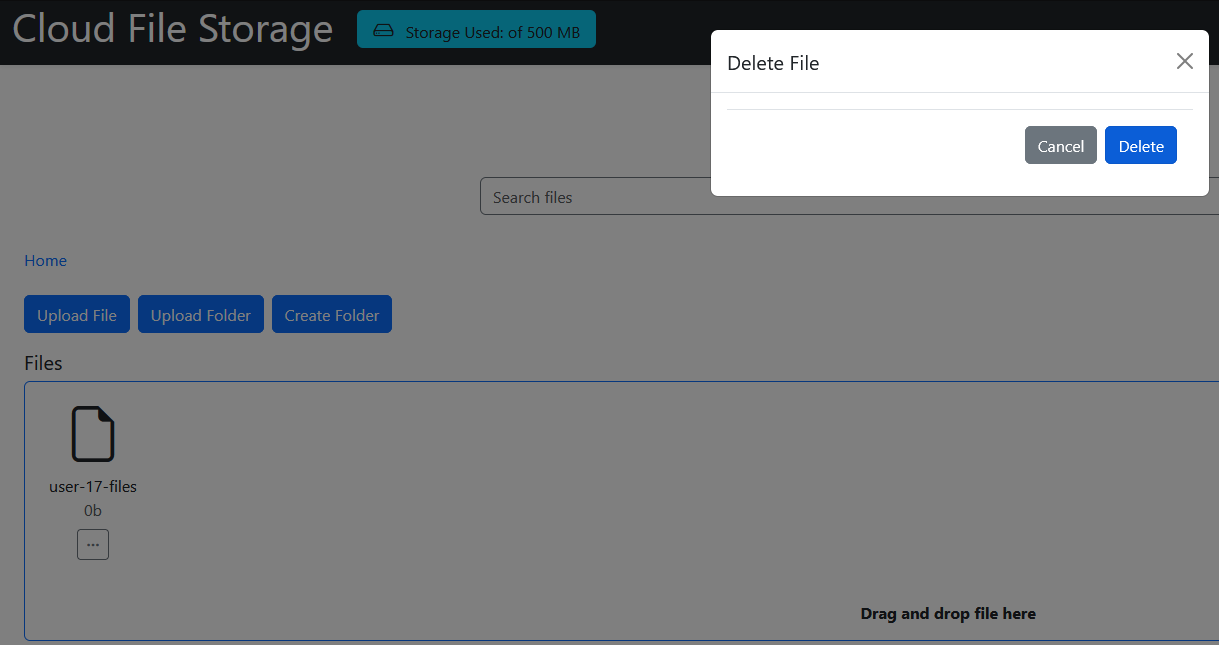


Рисунок 12 – Ошибка с удалением файла

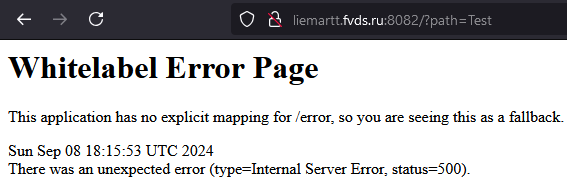


Рисунок 13 – Ошибка удаления всех файлов из директории

**Итог:** В результате тестирования были выявлены некоторые ошибки, которые препятствуют корректной работе системы в соответствии с техническим заданием. Проблемы возникают при выполнении ключевых функций, таких как обработка путей, создание и управление папками, загрузка и скачивание файлов, а также операции с облачным хранилищем. Эти ошибки требуют дальнейшей проработки и устранения для обеспечения стабильной работы системы

### Вывод

В результате тестирования программного продукта были выявлены 8 ошибок. Программный продукт не соответствует некоторым требованиям.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данной практической работы были изучены аспекты структуры технического задания для программного обеспечения. Также были приобретены практические навыки по созданию документации и написанию ТЗ для собственного продукта, а также тестирования программного продукта методом «Черного ящика».